

Parallel benchmarks for ARM processors in the high energy context

Joshua Wyatt Smith and Andrew Hamilton

University of Cape Town, Physics Department

E-mail: joshua.wyatt.smith@cern.ch

Abstract. High Performance Computing is relevant in many applications around the world, particularly high energy physics. Experiments such as ATLAS, CMS, ALICE and LHCb generate huge amounts of data which need to be stored and analyzed at server farms located on site at CERN and around the world. Apart from the initial cost of setting up an effective server farm the cost of power consumption and cooling are significant. The proposed solution to reduce costs without losing performance is to make use of ARM[®] processors found in nearly all smartphones and tablet computers. Their low power consumption, low cost and respectable processing speed makes them an interesting choice for future large scale parallel data processing centers. Benchmarks on the Cortex[™]-A series of ARM processors including the HPL and PMBW suites will be presented as well as preliminary results from the PROOF benchmark in the context of high energy physics will be analyzed.

1. Introduction

High energy physics (HEP) at the Large Hadron Collider (LHC) [1] creates an enormous amount of data that need to be stored for later analysis. Dedicated server farms have been built at CERN (Tier 0) as well as around the world (Tier 1's and Tier 2's) and are connected through The Worldwide LHC Computing Grid (WLCG). The initial setup costs of these server farms can be immense and the costs to maintain such a server farm can be even larger. Cooling and the power required to maintain such servers at their peak performance make server farms an expensive venture. In early 2013 Tier 0 had a power capacity of 3.5MW.

The proposed solution is to make use of ARM[®] processors [2] from here on referred to as ARM. These are found in smartphones and tablet computers where the combination of low power consumption and high performance is the top priority. Significant savings might be achieved if ARM processors are able to cope with the huge amount of data processed. This letter serves to explore the capabilities of ARM processors through parallel processing benchmarks.

2. Hardware and software

The ARM processor was designed to perform only a few instructions at once. This reduces the need for hardware such as transistors and thus minimizes power consumption. This letter addresses three system on chip (SoC) setups in the Cortex[™]-A range, namely the A7 MPCore[™], A9 MPCore[™] and A15 MPCore[™] [3, 4, 5], from here on referred to as the A7, A9 and A15 respectively. Table 1 summarizes the different processors used. An obvious advantage to the A9 is the number of cores, however this is irrelevant because the number of cores is an

intrinsic property of the SoC. FPU refers to *Floating Point Units*. The FPU generate results for the speed at which multiplication and addition operations are carried out. v3/4 refers to the respective FPU version. A traditional Intel[®] computer is also used (Hep405). This serves to provide some reference to the reader. Power measurements for the A7, A9 and A15 were taken using a Fluke 289 Digital Multimeter. Measurements on Hep405 were taken using the Intel[®] Power Gadget. However, if the power usage characteristics and temperatures vary from those used in Intel's calibration then there will be errors between estimated and actual power usage. For this reason, when referring to Hep405's results the reader must take care in remembering that these values serve as more of an estimate.

Table 1. The different setups with key features.

Setup	Processor	Cores	RAM	Cache	FPU	OS
Cubietruck	AllWinner A20, 1.2GHz	A7 dual core	2GiB DDR3	512 KiB L2	VFPv4	Archlinux, hard float
Wandboard-Quad	Freescall i.MX6 Quad, 996MHz	A9 quad core	2GiB DDR3	32KiB L1, 1 MiB L2	VFPv3	Archlinux, hard float
ArndaleBoard-K	Samsung Exynos 5250, 1.7GHz	A15 dual core	2GiB DDR3	32 KiB L1, 1MiB L2	VFPv4	Fedora 19, hard float
Hep405	Intel [®] Core i7-2600, 3.4GHz	quad core	16GiB DDR3	256KiB L1, 1MiB L2, 8MiB L3	-	Scientific Linux 6

3. Benchmarks and results

3.1. High Performance LINPACK suite

The LINPACK benchmark [6] is historically one of the most common tests in high performance computing (HPC) being used as early as the 1980's. The High-Performance LINPACK (HPL) benchmark is the parallel version of the LINPACK benchmark which is used to rank the world's TOP500 supercomputers. The user can specify how much memory to commit to solving the largest problem that the machine is capable of solving. It calculates the floating point operations per second or *flops* of a system by splitting the large matrix into blocks that are then solved on different cores or CPUs. This enables several blocks to be worked on in parallel. Ideally the increase in speed is scalable, i.e. four cores is four times quicker than one core, however, communication and latency between the cores hampers the performance and so speedup is never actually 100%. A list of block sizes and matrix sizes are specified at the beginning of the run (in ascending order). Then each matrix size is iterated over using the increasing block sizes. Thus, there is an overall increase in transferred matrix size from left to right.

Figure 1 shows the power consumption with the respective *Gflops* and *Gflops/watt*. Power measurements were taken at a 7 second resolution. The grey area is an "envelope function" which splits the data into blocks along the time axis and then finds the average power value for that window. It gives a representation of the average power consumption which is indicated by the blue line. A trend we see for the A7 and A9 is that as the size of the matrix block that is passed to the separate cores increases, the average power consumption decreases. This is because slower computation means that less work in the form of communication has to happen between the processors. The average power consumption remains fairly constant with the A15 because of the faster clock speed of 1.7GHz. The *Gflops/watt* is calculated by taking the average power consumption for the time window where there is a HPL measurement. We can see that the A9 is

the most efficient provided the block size is relatively large. The A7 does not have a fast enough processor and the A15 uses too much power for the output it delivers. The A9's *Gflops/watt* is comparable to the output of Hep405.

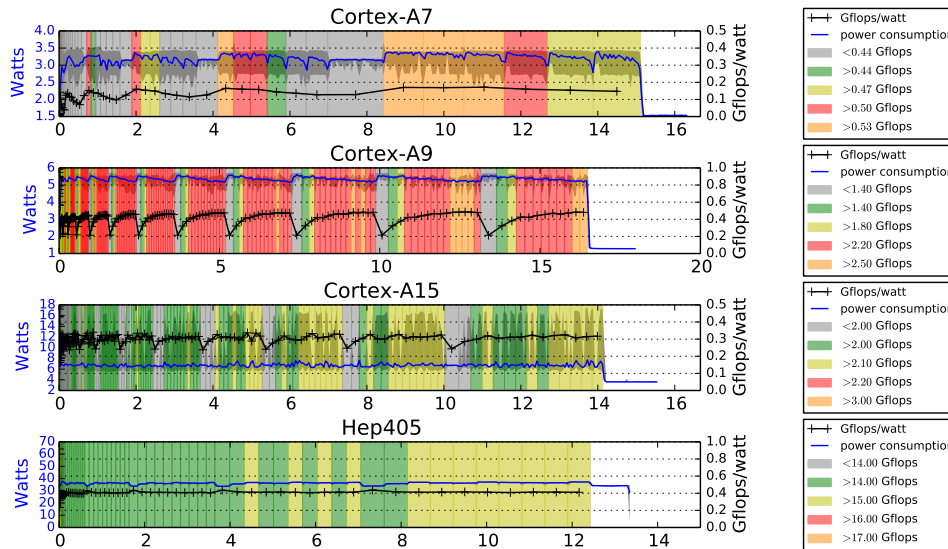


Figure 1. The relationship between matrix size, block size, *Gflops*, power consumption and *Gflops/watt* for the HPL benchmark. All *Gflops* values in each legend are present in each graph, they just have too small time scales to be seen.

3.2. Parallel Memory Bandwidth Benchmark suite

The Parallel Memory Bandwidth Benchmark (PMBW) [7] is a relatively new suite that measures bandwidth capabilities of a multi-core computer. This is an important test because more cores result in the floating point performance increasing in a linear fashion. However, if the memory bandwidth is not capable of processing the data fast enough those processors will stall. Unlike floating point units the memory bandwidth does not scale with the number of cores running in parallel. The code was developed in assembler language which means compile flags for the SoC become unimportant, thus there are no optimizations. The code uses two general synthetic access patterns, namely sequential scanning and pure random access. A real world application will fit somewhere in-between the two tests. The results for the benchmark are plotted in Figure 2.

For single threads, the A15 performs better than the A9 and A7 as expected due to the higher clock speed. If we look at the SoC with threads equal to the number of processors, for small cache memory transfers the peak bandwidth for the A9 and A15 are comparable but as the array size increases and bandwidth starts to stabilize the A9 doesn't perform as well as the A7 and A15 due to the lower clock speed at 996MHz. This figure shows only the results for the tests in which 32 bit message sizes are transferred for each clock cycle. It needs to be mentioned that the A15 has a higher peak performance when transferring 64 bit message sizes but it cannot maintain these speeds. As the total array size increases the bandwidth drops below the 32 bit values. For reference, on a single thread Hep405 is able to reach a peak bandwidth of 121 GiB/s for a 256 bit message size while reading and 60GiB/s while writing. It reaches 298GiB/s for reading and 163GiB/s for writing when running 4 threads. It is accurate to say that none of the ARM processors perform very well when it comes to memory bandwidth. A solution relies on using 64-bit architecture for the SoC instead of 32-bits. This will improve bandwidth speed

as it allows wider memory registers and access to more RAM. These chips have been released recently, however it will still be some time before they are put onto development boards.

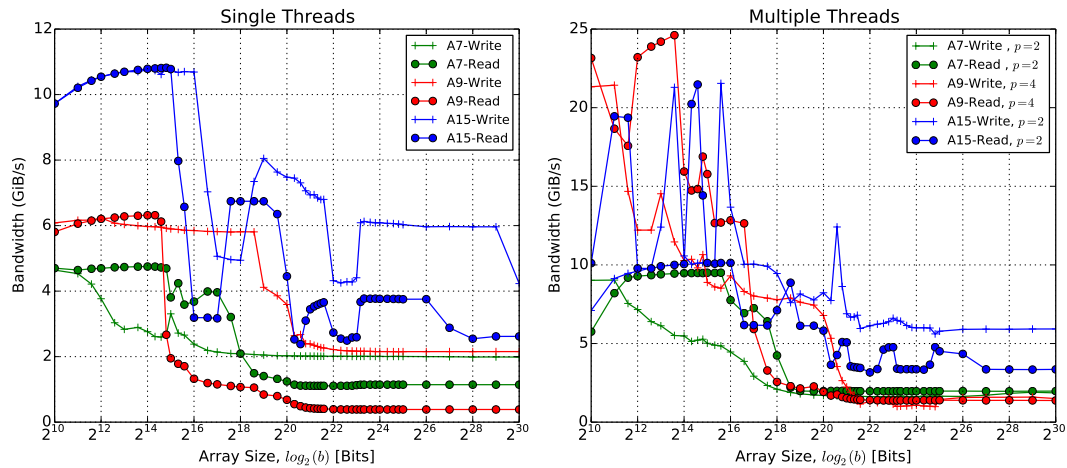


Figure 2. Memory bandwidth results for each processor. Read/Write refers to scanning operations (opposed to permutation operations) performed while doing read or write tasks. Each routine transfers 32-bits with increasing total array size.

3.3. PROOF benchmark suite

PROOF (Parallel ROOT Facility) [8], is a parallel extension to the well known data analysis tool used in HEP called ROOT [9]. It was designed as an extensive test suite in benchmarking potential configurations and performances of multicore computer clusters. PROOF exploits the fact that data analysis in the HEP context can be easily parallelizable. Tasks can therefore be neatly split up onto the individual cores (or *workers*) of each computer.¹ If the task size increases in proportion to the number of processors then the results are scalable. However, if data gets distributed or read sequentially by one core this may cause bottlenecks in the memory. Thus, different topologies can be tested such as having master nodes within smaller clusters which further facilitate communication. The CPU benchmark starts off by performing measurements with 1 active *worker* and enables additional *workers* at the start of each test. This benchmark is more appropriate for a setup with more nodes and cores, however it also returns a value for the SoC as well as a normalized value for a single *worker* which is good for comparing each of the ARM processors. The default setting creates 16 1d histograms filled with $1000000 \times \text{workers}$ random numbers. MRGPS is Mega Random Generations Per Second. One random generation produces 16 gaussian numbers and fills 16 histograms. The output is given for the SoC, as well as the normalized performance per worker. Power consumption was recorded with a resolution of 1 second. The results are shown in Table 2.

For the ARM processors the A9 is the most efficient returning the largest value for MRGPS and MRGPS/watt. However, it performs significantly worse than Hep405. This is a problem for the ARM processors as this is an easily parallelizable benchmark and the results should be very close to scaling linearly. We don't want to sacrifice speed and so it means that in order to reach computation speeds similar to that of Hep405 a minimum of 19 A9's must be present. This doesn't reduce power consumption.

¹ The number of *workers* isn't necessarily the same as the number of cores. For this letter, results showing *workers* equal to the number of cores were chosen.

Table 2. The PROOF CPU benchmark.

Setup	Power (W)	Workers	MRGPS	$\frac{MRGPS}{worker}$	$\frac{MRGPS}{watt}$
A7	1.997	2	0.108	0.055	0.054
A9	2.999	4	0.301	0.075	0.100
A15	7.254	2	0.296	0.118	0.041
Hep405	31.766	4	5.525	1.347	0.174

4. Conclusions

Table 3 contains a summary of the results. The A7, A9 and A15 use significantly less power than traditional processors. However, they are also significantly slower and so parallel computing for the right type of problem must be exploited. From a performance point of view the dual core A15 is only barely faster than the quad core A9. In terms of performance per watt the A9 performs better. The question arises as to whether or not it is beneficial to replace current computer processors with ARM processors. From our findings there is little advantage (if any) in using the 32-bit ARM processors presented in this letter. There is however, a quad core A15 available as well as the recently released CortexTM-A50 range with power efficient 64-bit architecture. Development boards with this newer technology should be available soon and may have a large impact on processing speeds and memory bandwidth.

Table 3. Summary of the important results for each benchmark.

	A7	A9	A15	Intel [®] i7
Cores	2	4	2	4
Idle (W)	1.518	1.282	3.573	4.356
HPL: Average <i>Gflops</i> /W	0.109	0.408	0.323	0.421
PMBW: Max read (GiB/s)	9.505	24.611	21.472	298.264
PMBW: Max write (GiB/s)	9.035	21.428	21.546	163.123
PROOF: $\frac{MRGPS}{watt}$	0.054	0.100	0.041	0.174

References

- [1] *Journal of Instrumentation*, vol. 3. IOP and SISSA, 2008.
- [2] ARM, *Procedure Call Standard for the ARM[®] Architecture*. ARM Ltd, November, 2012.
- [3] ARM Ltd, *CortexTM-A7 MPCoreTM Technical Reference Manual*, r0p5 ed., April, 2013.
- [4] ARM Ltd, *CortexTM-A9 MPCoreTM Technical Reference Manual*, r4p1 ed., June, 2012.
- [5] ARM Ltd, *CortexTM-A15 MPCoreTM Processor Technical Reference Manual*, r4p0 ed., June, 2013.
- [6] J. Dongarra, *Linpack benchmark*, Computer Science Technical Report CS - 89 - 85, University of Tennessee.
- [7] T. Bingmann, “pmbw - Parallel Memory Bandwidth Benchmark / Measurement.” website, 2013.
- [8] ACAT, *The PROOF benchmark suite measuring PROOF performance*, vol. Journal of Physics: Conference Series 368 (2012) 012020, IOP Publishing, 2011.
- [9] Antcheva, I. et al, ed., *ROOT - A C++ framework for petabyte data storage, statistical analysis and visualization*, vol. 180, 12; 2499-2512, Computer Physics Communications, 2009.